IST-ASM Retake Exam — 1st December 2022

Name:

- First, write your name in the box above. Then, have a quick read through all 5 exercises.
- In the end, you will write up your answers on this paper.
  - But please make a draft elsewhere first. Only hand in something readable.
- This is an open-book open-laptop exam: you may work on scrap paper or on your screen.
- Each question is independent from others.

Question 1 Perform the binary addition $-43+50$ in two's complement on 7 bits: convert both numbers to (signed) binary, then compute the sum on 7 bits. Show the details of your work, especially carry bits.

au moins deux points sur le -43 correct en binaire.

```
  1 1 1
    1 0 1 0 1 0 1
+   0 1 1 0 0 1 0
-----------------
    0 0 0 0 1 1 1 = 7
```

Question 2 The code below implements a certain mathematical function f: from two integers A and B, it computes $C = f(A, B)$. Give a simple expression for f.

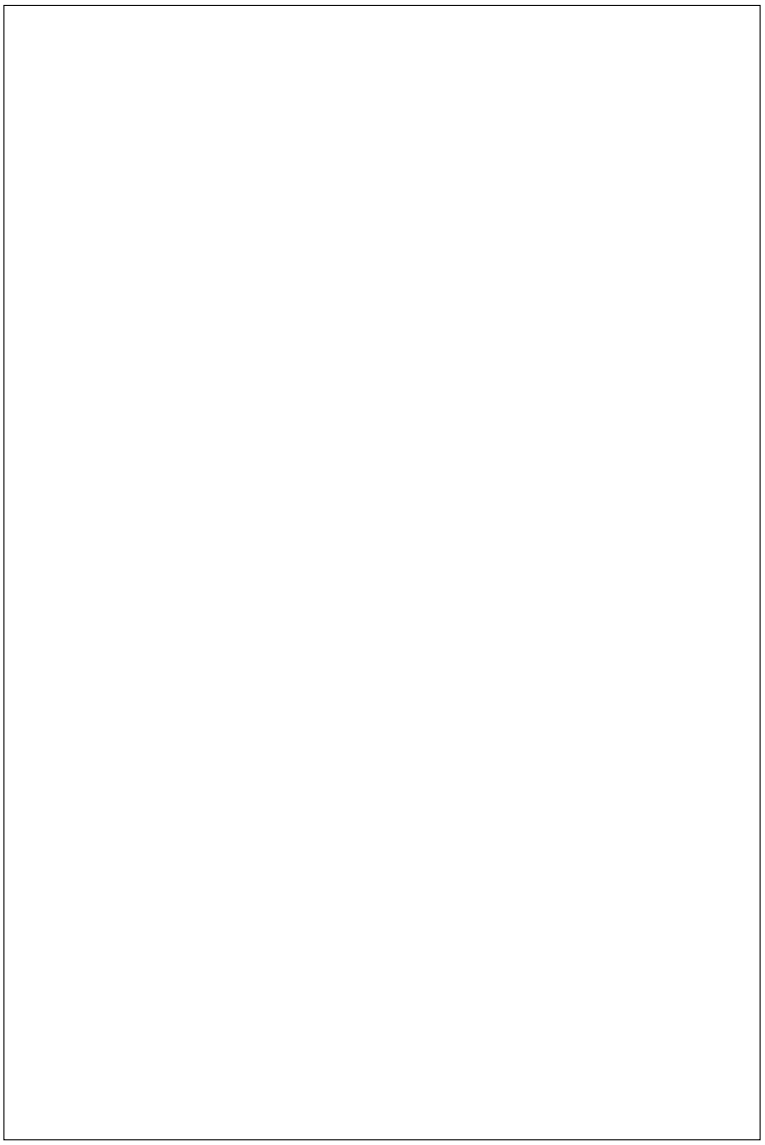
$f(A, B) =$

```
A: .word ...
B: .word ...
C: .word ...
main:
    load R1, [A]
    load R2, [B]
    mul R3, R1, R1
    mul R4, R2, R2
    add R3, R3, R4
    mul R4, R1, R2
    add R4, R4, R4
    add R1, R3, R4
    store [C], R1

    bra +0
```

f(A,B) = $A^2 + B^2 + 2AB = (A + B)^2$

Question 3 Write a program which computes the sum of the squares of the first N positive integers. For instance, with $N = 7$ you should find $1\times1 + 2\times2 + 3\times3 + 4\times4 + 5\times5 + 6\times6 + 7\times7 = 140$. Initially N is stored in R1, and at the end the result should be stored in R2.



```
        bra main

A:      .word 7

main:
        load R1, [A]
        leti R2, 0
loop:
        mul R3, r1, r1
        add r2, r2, r3
        dec r1
        bgtz r1, loop

        bra +0
```

Question 4 Write a program that loops over an array of numbers and finds both the maximum and minimum values. The length of the array is a (known) constant, as illustrated below.

```
T:   .word 13, 18, 5, 3, 10, 8, 20, 1, 14, 6
len: .word 10

main:
```

```
bra main

T:   .word 13, 18, 5, 3, 10, 8, 20, 1, 14, 6
len: .word 10

main:
        leti R1, 0 ;; index in T
        load R2, [len]

        leti r3, 0x80000000 ;; current max: INTMIN
        leti r4, 0x7FFFFFF  ;; current min: INTMAX

loop:
        muli R5, R1, 4
```

```
        leti R6, T
        add R5, R5, R6

        load R5, [R5] ;; T[i]
        bgt R3, R5, +8
        mov R3, R5

        blt R4, R5, +8
        mov R4, R5

        inc R1
        blt R1, R2, loop

        bra +0
```

Question 5 Definition: Given a pair of positive integers n and k such that $n \geqslant k \geqslant 0$, we define their *binomial coefficient* as the number of different k-element subsets of a fixed n-element set. This number is usually written $\binom{n}{k}$ and is read as "n choose k". For example, $\binom{4}{2} = 6$ because there are 6 ways to choose 2 elements from a 4-element set $\{a, b, c, d\}$: the different subsets are $\{a, b\}$, $\{a, c\}$, $\{a, d\}$, $\{b, c\}$, $\{b, d\}$, and $\{c, d\}$.

In this exercise, we are interested in the fact that there exists a recursive formula to compute these coefficients:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

The base case of the recursion is the fact that for any integer $n \geqslant 0$, we have $\binom{n}{n} = \binom{n}{0} = 1$.

Your task is to write a recursive `binomial` function which receives n and k in R1 and R2, respectively and returns $\binom{n}{k}$ in R1.

```
leti SP, 0x10000000
main:
    leti R1, 4
    leti R2, 2
    call binomial
    bra +0

binomial:
```

```
    leti SP, 0x10000000

main:
    leti R1, 4
    leti R2, 2
    call binomial
t1: ;; expect R1 == 6

    leti R1, 5
    leti R2, 3
    call binomial
t2: ;; expect R1 == 10

    leti R1, 8
    leti R2, 3
    call binomial
t3: ;; expect R1 == 56

    bra +0
```

```
    ;; input:  N in R1
    ;; input:  K in R2
    ;; output: (N choose K) in R1
binomial:
    push LR
    push R6
    push R5
    push R4
    push R3

    beq R1, R2, retone ; (N choose N)
    beq R2, R0, retone ; (N choose zero)

    mov R3, R1 ;; save N
    mov R4, R2 ;; save K

;; (N choose K) := (N-1 choose K-1) + (N-1 choose K)
    addi R1, R3, -1
    addi R2, R4, -1
    call binomial
    mov R5, R1 ;; save (N-1 choose K)

    addi R1, R3, -1
    mov  R2, R4
    call binomial
    mov R6, R1 ;; save (N-1 choose K-1)

    add R1, R5, R6
epilogue:
    pop R3
    pop R4
    pop R5
    pop R6
    pop LR
    ret

retone:
    leti R1, 1
    jmp epilogue
```